A Wide-Coverage Symbolic Natural Language Inference System

Jean-Philippe Bernardy
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
jean-philippe.bernardy@gu.se

Stergios Chatzikyriakidis
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
stergios.chatzikyriakidis@gu.se

Abstract

We present a system for Natural Language Inference which uses a dynamic semantics converter from abstract syntax trees to Coq types. It combines the fine-grainedness of a dynamic semantics system with the powerfulness of a state-of-the-art proof assistant. We evaluate the system on all sections of the FraCaS test suite, excluding section 6. This is the first system that does a complete run on the anaphora and ellipsis sections of the FraCaS. It has a better overall accuracy than any previous system.

1 Introduction

Natural Language Inference (NLI) is the task of determining of whether an NL hypothesis H follows from an NL premise(s) P. NLI has received a lot of attention in the Computational Semantics literature and has been approached using a variety of techniques, ranging from logical approaches (Bos, 2008; Mineshima et al., 2015; Abzianidze, 2015; Bernardy and Chatzikyriakidis, 2017), all the way to the recent Deep Learning (DL) models for NLI. The latter approaches, following a general trend in NLP, have been dominating NLI and a number of impressive results have been produced (Kim et al., 2018; Radford et al., 2018; Liu et al., 2019).[1] State-of-the-art DL systems achieve an accuracy of around 0.9 when tested on suitable datasets. However, the datasets that are used are assuming a definition of inference that can be thought to be 'looser' or less precise compared to the definition assumed in platforms based in logical approaches (Bernardy and Chatzikyriakidis, 2019). For example, consider the following example from the SNLI dataset, predominatly used to test DL approaches:

(1) **P** A man selling donuts to a customer during a world exhibition event held in the city of Angeles.
H A woman drinks her coffee in a small cafe.
Label: Contradiction [SNLI]

In (1), a number of non-trivial assumptions have to be made in order to arrive at a contradiction: a) the two situations described have to be taken to refer to the same situation in order to judge that the latter contradicts the former, b) the indefinite article in the premise has to be identified with the indefinite article in the hypothesis. (Additionally considering that a person cannot be a man selling donuts and a woman drinking coffee at the same time.) While this can be part of the reasoning humans perform, it is not the only possibility. More precise, logical reasoning is also a possibility, and will render the above label as unknown. Furthermore, reasoning can get very fine-grained as the Contained Deletion ellipsis example (2) below shows:

(2) **P1** Bill spoke to everyone that John did [elliptic V2].
P2 John spoke to Mary.
Q Did Bill speak to Mary?
H Bill spoke to Mary.
Label: Yes [FraCas 173]

For this reason, and despite the dominance of DL approaches in pretty much all NLP tasks, logical approaches continue to be developed and evaluated on datasets like the FraCaS test suite and the SICK dataset (Marelli et al., 2014). Bernardy and Chatzikyriakidis (2017) define a correspondence between abstract syntax parse trees of the FraCas examples, parsed using the Grammatical Framework (GF, Ranta (2011)), and modern type-theoretic semantics that are output in the Coq proof assistant (the FraCoq system). The accuracy is 0.85 for 5 sections of the FraCaS test suite.

[1]These are the three systems with the best results on SNLI in increasing order at the time of writing.

The LANGPRO system presented by Abzianidze (2015) is based on a Natural Logic tableau theorem prover. It achieves an accuracy of .82 on the SICK dataset.

In this paper, we concentrate on this sort of fine-grained, logical reasoning. In particular, we present a logic-based system that deals with many linguistic phenomena *at the same time*. It is the first system covering the sections on ellipsis and anaphora in the FraCaS test suite and has the best coverage and accuracy on the overall test suite.

2 Background

GF In GF, abstract syntax is comprised of: a) a number of syntactic categories, and b) a number of syntactic construction functions. The latter provide the means to compose basic syntactic categories into more complex ones. For example, consider the constructor: $AdjCN : AP \to CN \to CN$. This expresses that one can append an adjectival phrase to a common noun and obtain a new common noun. Furthermore, GF is equipped with a library of mappings from abstract syntax to the concrete syntax of various natural languages. These mappings can be inverted by GF, thus offering parsing from natural text into abstract syntax. However, in this project we skip the parsing phase and use the parse trees constructed by Ljunglöf and Siverbo (2011), thereby avoiding any syntactic ambiguity.

Coq Coq is an interactive theorem prover (proof assistant) based on the calculus of inductive constructions (CiC), i.e. a lambda calculus with dependent types. Coq is a very powerful reasoning engine that makes it fit for the task of NLI, when the latter is formalized as a theorem proving task. It supports notably dependent typing and subtyping, which are instrumental in expressing NL semantics.

Dynamic Monadic Semantics Dynamic Monadic Semantics have been proven to be an effective way of dealing with anaphora and ellipsis. There are a number of approaches using monads or other equivalent constructions (e.g. continuations as in the work of de Groote (2006)) for anaphora and ellipsis Shan (2002); Unger (2011); Barker and chieh Shan (2004); Qian et al. (2016); Charlow (2017). In this paper, we follow the approach described in Bernardy et al.. More details are given in the next section.

3 Overview of the system

Our system consists of two main parts.

1. A converter from syntax trees to types. The syntax trees follow the GF formalism, and the types follow the Coq formalism. The converter itself is a Haskell Program, which implements a dynamic semantics and comprises the bulk of our system.
2. A number of type-theoretical combinators, that encode semantical aspects which have no influence on the dynamic part. Such aspects include the treatment of adjectives (intersective, subsective, etc.) and adverbs (veridical or not).

The architecture is represented schematically in Figure 1.

All the underlying systems (GF, Haskell, Coq) are based on lambda calculi with types. We take advantage of typing, ensuring that each translation preserve typing, locally:

1. Every GF syntactic category C is mapped to a type noted $[\![C]\!]$.
2. GF Functional types are mapped compositionally : $[\![A \to B]\!] = [\![A]\!] \to [\![B]\!]$
3. Every GF syntactic construction function ($f : X$) is mapped to a function $[\![f]\!]$ such that $[\![f]\!] : [\![X]\!]$.
4. GF function applications are mapped compositionally: $[\![t(u)]\!] = [\![t]\!]([\![u]\!])$.

Because all systems embed the simply-typed lambda calculus, ensuring type-preservation locally means that types are preserved globally. Therefore, we are certain that every GF syntax tree can be mapped to Haskell, and eventually Coq, without error.

The dynamic semantics follows a monadic structure, as pioneered by Shan (2002). There are two kinds of effects carried by the monad. The first one comprises a series of updates and queries of stateful elements. There is one piece of updateable state for every element which can be referred to by anaphoric expressions. These can be the usual ones (like NPs), but also less usual ones (like 2-place verbs, or a quantity — which we illustrate below). The other kind of effects is *non-determinism.* We use non-determinism to model the property that linguistic expressions can have several interpretations. The monadic structure allows to locally express that a given expression has

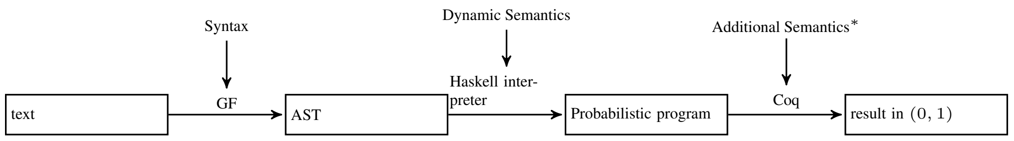

Figure 1: Phases in our system. (∗) At the level of Coq, we handle the details of the adverbial (veridicality properties) and adjectival semantics (division into subsective, extentional, non-committal, etc. categories.)

several meanings; the monadic bind ensures that all combinations of meanings are considered at the top-level, combinatorially. This dynamic semantics allows us to model many phenomena in a precise way.

Anaphora Thanks to the above system, we can handle many anaphoric cases, including E-Type and Donkey anaphora. Indeed, even objects which have no syntactic representation can be added to the environment. We follow here the general monadic semantics approach as outlined by Unger (2011). However, we use a more general scope-extension mechanism, which allows us to support examples like the following:

(3) **P1** Every committee has a chairman.
P2 He is appointed its members.
H Every committee has a chairman appointed by members of the committee.
Label: YES [FraCaS 122]

In the above example, the pronoun "he" is allowed to refer to the object quantified over by "every", whose scope is extended accordingly. We describe the anaphora resolution system in every detail in a manuscript (Bernardy et al.).

Ellipsis Ellipsis is handled in essentially the same way as anaphora. This method is made especially straightforward thanks to using GF syntax trees, which require an explicit argument for each predicate. Thus, ellipsis are made explicit by the parsing phase. Such ellptic expressions are handled in the same way as anaphora. For example, in (2) repeated below as (4), the argument of "did" is explicitly marked as an elliptic V2, which we resolve to "speak" in that context:

(4) **P1** Bill spoke to everyone that John did [elliptic V2].
P2 John spoke to Mary.
Q Did Bill speak to Mary?
H Bill spoke to Mary.
Label: Yes [FraCas 173]

Definites A naive way to handle definites is using an existential type. However, if the semantics does not feature a dynamic element, then the existential quantification is introduced locally. This means that the quantifier can be introduced in the wrong Context. Consider the phrase "everyone pets the dog". The structure of the interpretation would be $\forall x.person(x) \rightarrow \exists y.dog(y) \wedge pet(x,y)$. Instead, our take is that definites should be treated as an anaphoric expression with an implicit referent. That is, if the referent is not found in the discourse, then it will be forcibly introduced, using an existential type, *at the top-level* of the expression. To be able to do this, we record all definites without referent, using another portion of the environment (using a monadic effect). For the above example, we obtain the desired interpretation: $\exists y.dog(y) \wedge (\forall x.person(x) \rightarrow pet(x,y))$.

Phrasal comparatives Previous attempts to tackle the section of the FraCaS test suite devoted to comparatives showed that handling them is not easy. Our strategy here is to leverage our dynamic semantics, revealing an anaphoric element of comparatives. Indeed, consider the hypothesis of (FraCaS 239): "ITEL won more orders than APCOM lost." We postulate that this sentence is equivalent to the following two separate parts: "APCOM lost zero or more orders. ITEL won more orders [than some elliptic quantity]." A quantity is introduced every time we talk about some quantity (indexed by a CN, in this case "orders"), and it can be referred to by a comparative, later in the discourse. Using this idea, we can go one level deeper in the interpretation of our example: "APCOM lost θ orders. $\theta \geq 0$. ITEL won at least θ+1 orders.". We see here how the quantities are introduced. They are added to the environment so that, they can be referred to as elliptic quantity expressions.[2] Fi-

[2]The degree parameter assumption is not new in the formal semantics literature (Cresswell, 1976; Heim, 2000; Kennedy, 2007; Chatzikyriakidis and Luo, 2017) among many others. The specific details and computational imple-

nally, “more” is systematically intepreted as “at least ¡elliptic quantity¿+1”. This treatment, which we illustrated here on an example, is systematic in our implementation.

Adjectives We interpret gradable adjectives using a pair of a measure $m : objects \rightarrow Z$ and a threshold $\tau : Z$, where Z is treated as an abstract ordered ring by Coq. (This structure has no dynamic aspect in our model, and thus is entirely handled within Coq.) For subsective adjectives, τ will additionally depend on the class of the object in question. This structure has the benefit that opposite adjectives can be easily represented (measures are opposites $\forall x.m_1 x = \neg m_2 x$ and thresholds do not overlap $\tau_1 + \tau_2 > 0$). Formalization aside, this idea is reminiscent of degree-based approaches to gradable adjectives of Cresswell (1976); Kennedy (2007). Additionally adjectival predicates, as present in the FraCaS suite, are interpreted as linear inequations in Z. Solving systems of such inequations is decidable. Indeed, the tactic that Coq offers for this purpose can solve all such problems in the FraCaS suite, automatically.

Adverbs Another point, of minor theoretical importance but major practical one, is our handling of adverbial phrases. We interpret adverbs (and in general all adverbial and prepositional phrases) as VP-modifiers: $Adv = VP \rightarrow VP$, where $VP = object \rightarrow Prop$. However, applying adverbs to verb-phrases heavily complicates the Coq proofs, because such phrases can contain quantifiers. Therefore, we instead move the adverbs, so that they apply to (atomic) verbs only. Proofs can then be simplify accordingly.

4 Results and evaluation

We evaluated FraCoq against 8 sections of the FraCaS test suite, for a total of 259 cases. We excluded only section 7, “temporal reference”. The reason for doing so is that, in our view, it contains too many examples which require *ad-hoc* treatment, and thus makes little sense to include without complementing it with a more thorough suite which captures a more complete landscape of the phenomena that section 7 touches.

FraCaS classifies each problem as either entailment (YES), entailment of the opposite (NO) or no entailment (UNK). In this work, we have amended the FraCaS suite to correct a few problems. First,

mentation, however, are.

test case	new class	comment
005	UNK	missing hypothesis: there are italian tenors
056	Yes	Already identified as such by MacCartney
069	Unk	Mary could have used someone else’s workstation
119	Unk	*ibid.*
181	Yes	for the same reason as 180
226	Yes	

Table 1: Overruled FraCaS cases

certain test case are not formed correctly. Those were already identified by MacCartney and Manning (2007) as such (using an “undef” labelling), and we removed those. Second, a few test cases occur twice in the suite, but with two different labellings (one YES and one UNK), with an annotation that those labellings correspond to different readings. However, elsewhere in the suite, if a problem has several readings but only one has entailment, it occurs only once and is marked as YES. To make the test suite consistent, if one reading yields entailment we have always considered it as YES. We have also removed case 199 (which appears to be vacuous). Finally we changed the labelling of 6 cases which appeared to have been misclassified. We note that the majority of the mistaken classifications occur in sections 3 and 4, which have not been previously attempted and thus, we propose, have not been properly scrutinized. In terms of comparison, this only has a minor effect, since our system is the first system to run sections 3 and 4.

Our system classifies a case as YES if a proof can be constructed from the premises to the hypothesis, NO if a proof of the negated hypothesis can be constructed and UNK otherwise. Because we work with a non-decidable logic, one cannot *in general* conclude decisively that no proof exists. Thus, we consider here that no proof exists if it cannot be constructed with reasonable effort. In particular, we test at the minimum that the automatic proof search built in Coq does not succeed before classifying a problem as UNK.[3]

Table 2 shows a considerable improvement over earlier approaches in terms of coverage, with three more sections covered over previous approaches. We thus cover 259 out of 337 cases (77%), compared to at most 174 cases (52%) in previous work. Additionally, our system performs generally the

[3]The other way this can be done is by introducing a time-out as Mineshima et al. (2015) have done.

Section	#cases	Ours	FC	MINE	Nut	Langpro
Quantifiers	75	.96	.96	.77	.53	.93
		74				44
Plurals	33	.82	.76	.67	.52	.73
						24
Anaphora	28	.86	-	-	-	-
Ellipsis	52	.87	-	-	-	-
Adjectives	22	.95	.95	.68	.32	.73
		20				12
Comparatives	31	.87	.56	.48	.45	-
Temporal	75	-	-	-	-	-
Verbs	8	.75	-	-	-	-
Attitudes	13	.92	.85	.77	.46	.92
						9
Total	337	.89	.83	.69	.50	.85
		259	174	174	174	89

Table 2: Accuracy of our system compared to others. "Ours" refers to the approach presented in this paper. When a system does not handle the nominal number of test cases (shown in the second column), the actual number of test cases attempted is shown below the accuracy figure, in smaller font. "FraCoq" refers to the work of Bernardy and Chatzikyriakidis (2017). "MINE" refers to the approach of Mineshima et al. (2015), "NUT" to the CCG system that utilizes the first-order automated theorem prover *nutcracker* (Bos, 2008), and "Langpro" to the system presented by Abzianidze (2015). A dash indicates that no attempt was made for the section.

best in terms of accuracy. In particular, section 6 largely improves in accuracy, which we attribute to our dynamic semantics analysis of comparatives.

error analysis Our system fails to correctly classify 28 cases out of 259. We give here a summary of the missing features which are responsible for the failures. The biggest source of error is incomplete handling of group readings. (FraCaS 013, 014, 046, 084, 111, 124, 126, 127, 137, 171, 172, 191, 193, 195, 243, 250, 333, 346). These are cases where a syntactic conjunction of individuals is treated as a semantic group, or where precise counting of the members of a group is necessary. Other problematic cases include definite plurals with no universal readings (091, 094, 095). Additionally, neither measure phrases (242) nor attributive comparatives (244, 245) are handled.

5 Conclusions and Future Work

We presented a system converting GF trees to Coq types using dynamic semantics. The system outperforms the state of the art in logical approaches when tested on the FraCaS and is the only system to date to perform a run on the FraCaS ellipsis/anaphora section. The system is precise enough to form the start of a precise NL reasoner for controlled domains. In the future, we plan to extend the system to cover the remaining section of the FraCaS (tense/aspect), and also develop a more applied version to perform reasoning on controlled NL domains.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of EMNLP15*.

Chris Barker and Chung chieh Shan. 2004. Continuations in natural language. In *in Proceedings of the fourth ACM SIGPLAN workshop on continuations, Hayo Thielecke*, pages 55–64.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2017. A type-theoretical system for the fracas test suite: Grammatical framework meets coq. In *IWCS 2017-12th International Conference on Computational Semantics-Long papers*.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2019. What kind of natural language inference are nlp systems learning: Is this enough? In *Proceedings of ICAART*.

Jean-Philippe Bernardy, Stergios Chatzikyriakidis, and Aleksandre Maskharashvili. A computational treatment of anaphora and its algorithmic implementation. Manuscript available online `https://bit.ly/2xQ4G2M`.

Johan Bos. 2008. Wide-coverage semantic analysis with boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286. Association for Computational Linguistics.

Simon Charlow. 2017. A modular theory of pronouns and binding. In *Logic and Engineering of Natural Language Semantics (LENLS) 14*. Springer.

Stergios Chatzikyriakidis and Zhaohui Luo. 2017. Adjectival and adverbial modification: The view from modern type theories. *Journal of Logic, Language and Information*, 26(1):45–88.

Max J Cresswell. 1976. The semantics of degree. In *Montague grammar*, pages 261–292. Elsevier.

Philippe de Groote. 2006. Towards a montagovian account of dynamics. In *Semantics and Linguistic Theory*, volume 16, pages 1–16.

I. Heim. 2000. Degree operators and scope. In *Proceedings of SALT*, volume 10, pages 40–64.

Christopher Kennedy. 2007. Vagueness and grammar: The semantics of relative and absolute gradable adjectives. *Linguistics and philosophy*, 30(1):1–45.

Seonhoon Kim, Jin-Hyuk Hong, Inho Kang, and No-jun Kwak. 2018. Semantic sentence matching with densely-connected recurrent and co-attentive information. *arXiv preprint arXiv:1805.11360*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

P. Ljunglöf and M. Siverbo. 2011. A bilingual treebank for the FraCas test suite. Clt project report, University of Gothenburg.

Bill MacCartney and Christopher D Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200. Association for Computational Linguistics.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 1–8.

Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of EMNLP*.

Sai Qian, Philippe de Groote, and Maxime Amblard. 2016. Modal subordination in type theoretic dynamic logic. *LiLT (Linguistic Issues in Language Technology)*, 14.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Aarne Ranta. 2011. *Grammatical framework: Programming with multilingual grammars*. CSLI Publications.

Chung-chieh Shan. 2002. Monads for natural language semantics. *CoRR*, cs.CL/0205026.

Christina Unger. 2011. Dynamic semantics as monadic computation. In *JSAI International Symposium on Artificial Intelligence*, pages 68–81. Springer.